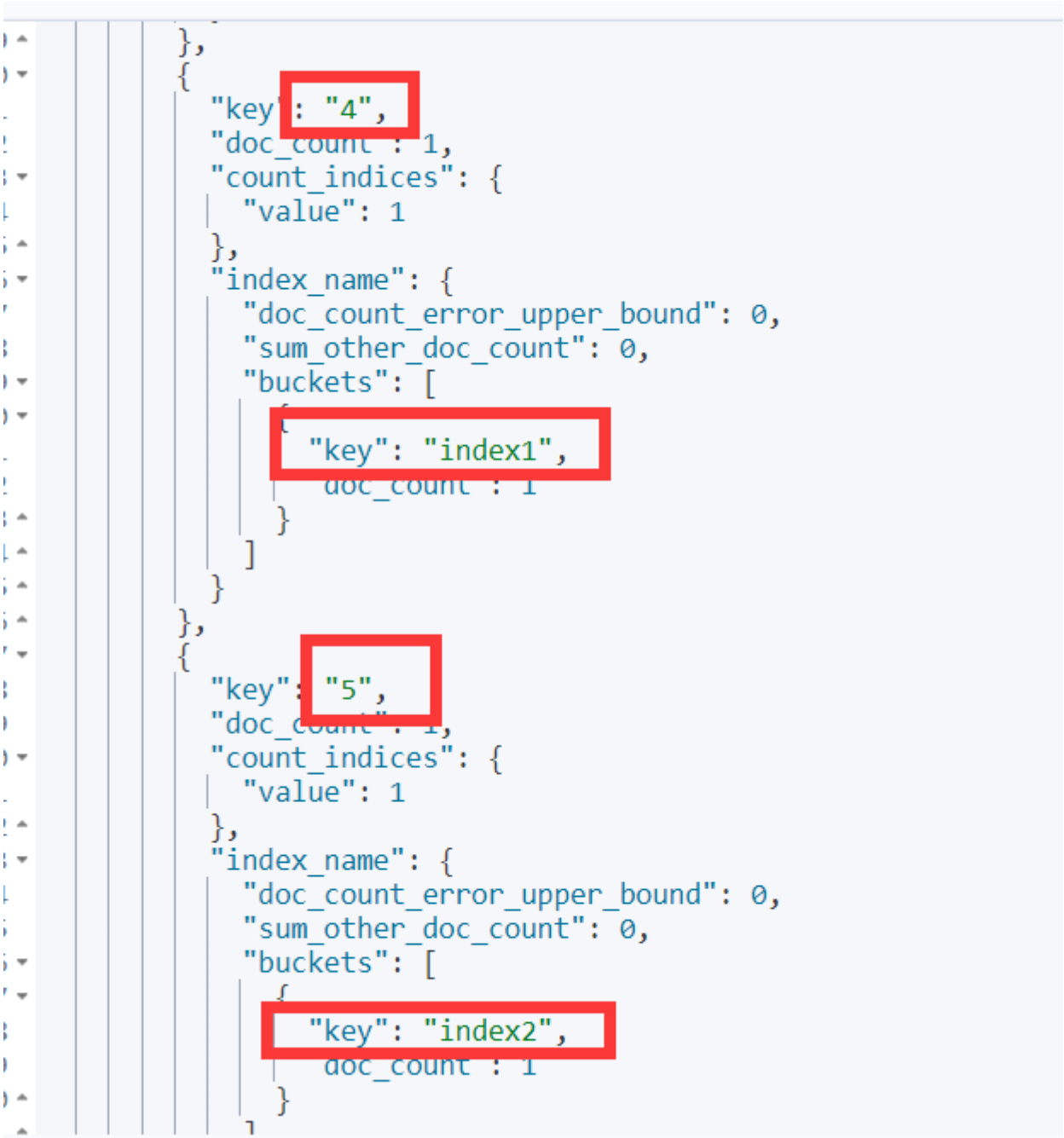
8月31日笔记_es复杂查询/聚合/过滤

过滤查询：http://www.360doc.com/content/22/0627/09/13042814_1037571558.shtml

代码：

```
//成功
GET index1,index2/_search
{
  "size": 0,
  "aggs": {
    "group_by_uid": {
      "terms": {
        "field": "name.keyword",
        "size": 1000000
      },
      "aggs": {
        "count_indices": {
          "cardinality": {
            "field": "_index"
          }
        },
        "values_bucket_filter_by_index_count": {
          "bucket_selector": {
            "buckets_path": {
              "count": "count_indices"
            },
            "script": "params.count < 2;"
          }
        },
        "index_name":{
          "terms": {
            "field": "_index",
            "size": 1
          }
        }
      }
    }
  }
}
```

结果：



合并多子句：

叶子子句：用以在将查询字符串与一个字段笔记

复合子句：用以合并其他的子句。bool子句允许合并其他合法子句，must，must_not should

过滤与查询

1、term过滤

用于精确匹配那些值，比如数字，日期，bool值或not_analyzed的字符串

2、terms过滤

terms允许多个匹配条件

3、range 允许安装指定范围查找数据

gt :: 大于

gte :: 大于等于

lt :: 小于

lte :: 小于等于

4、bool过滤

可以用来合并多个过滤条件结果的bool逻辑

must 多个查询条件的完全匹配，and

must_not 多个条件相反的匹配。

should 有一个条件匹配，相当于or

5、match_all查询

使用match_all可以查询到所有文档，是没有查询条件下的默认语句

```
{ "match_all": {} }
```

6、match查询

7、multi_match

在match基础上的多个搜索字段

8、bool查询

bool 查询与 bool 过滤相似，用于合并多个查询子句。不同的是， bool 过滤可以直接给

出是否匹配成功，

而 bool 查询要计算每一个查询子句的 _score （相关性分值）。

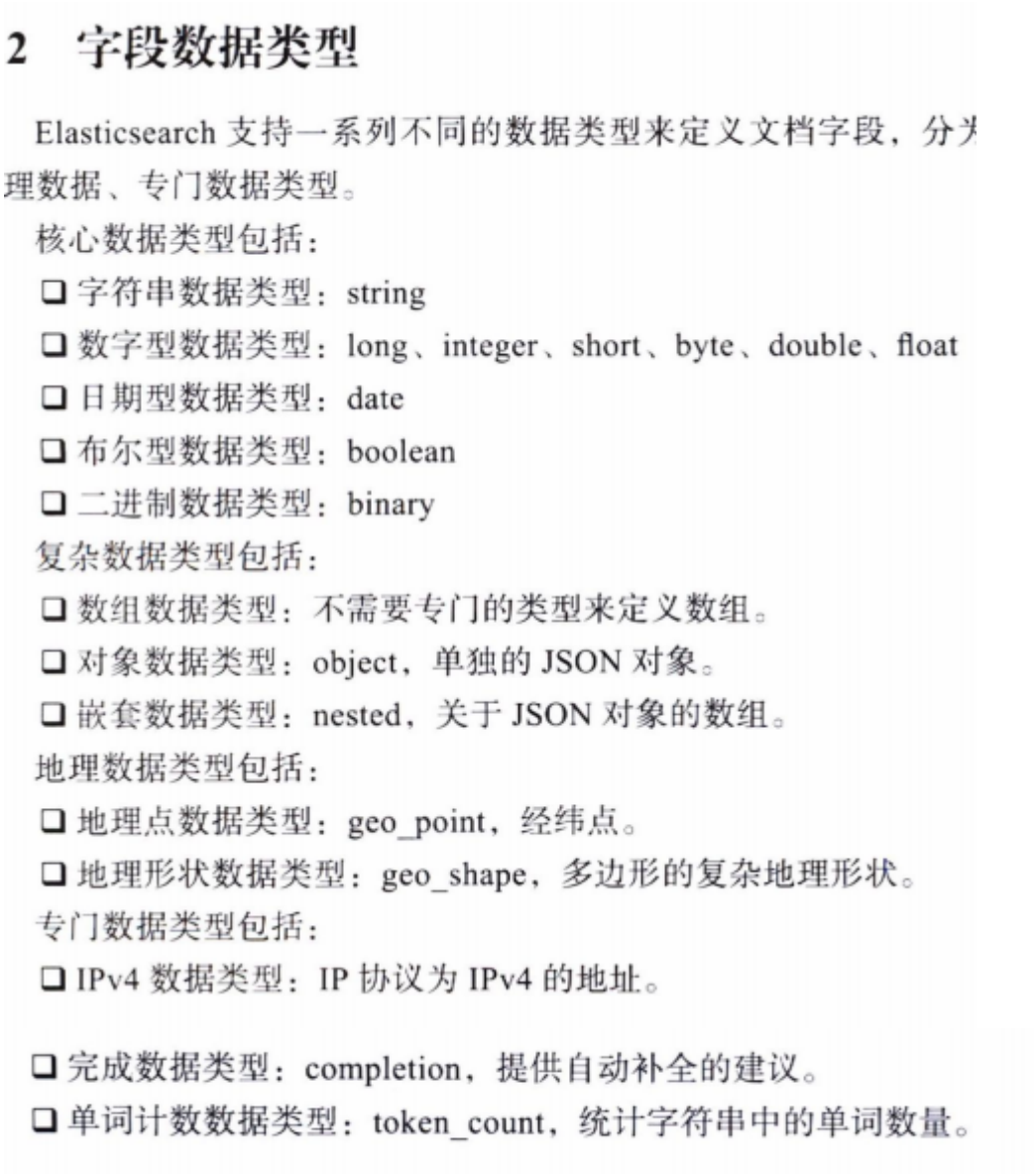
是一个标准查询，不管需要全文查询还是精确查询，基本上都要用它。如果使用match查询一个全文吧字段，它会在真正查询之前用分析器先分析match一下查询字符

```
//分页
GET /_search { "from": 30, "size": 10 }
```

多索引和多类别

映射

映射是定义存储结构和索引的文档类型以及字段的过程。索引中的每一个文档都有一个类型，每种类型都有它自己的映射。一个映射定义了文档结构内每个字段的数据结构类型。映射通过配置来定义字日期类型等等，或者文档中所有字段的值是否被_all字段索引等。



2 字段数据类型

Elasticsearch 支持一系列不同的数据类型来定义文档字段，分为
理数据、专门数据类型。

核心数据类型包括：

- 字符串数据类型：string
- 数字型数据类型：long、integer、short、byte、double、float
- 日期型数据类型：date
- 布尔型数据类型：boolean
- 二进制数据类型：binary

复杂数据类型包括：

- 数组数据类型：不需要专门的类型来定义数组。
- 对象数据类型：object，单独的 JSON 对象。
- 嵌套数据类型：nested，关于 JSON 对象的数组。

地理数据类型包括：

- 地理点数据类型：geo_point，经纬点。
- 地理形状数据类型：geo_shape，多边形的复杂地理形状。

专门数据类型包括：

- IPv4 数据类型：IP 协议为 IPv4 的地址。
- 完成数据类型：completion，提供自动补全的建议。
- 单词计数数据类型：token_count，统计字符串中的单词数量。

字符串数据类型：

- 全文本。全文本值通常用于基于文本的相关性搜索，全文本字段可以分词，即在索引执行之前通过一个分词器将字符串转换为单词列表。分词操作使得Elasticsearch可以在全文本字段上搜索单词。全文本字段不可用于排序，而且很少用于聚合。

元数据

每个文档都有与之关联的元数据，元字段是为了保证系统正常运转的内置字段。_index表示索引字段，_type表示映射字段，_id表示文档主键字段，这些字段都是以下划线开头的。当映射类被创建的时候，可以自定义一些元字段的行为。

表 3-12　标识元字段

参数	说明
_index	文档所属的索引
_uid	包含 _type 和 _id 的混合字段
_type	文档的映射类型
_id	文档的 ID

表 3-13　文档来源元字段

参数	说明
_source	作为文档内容的原始 JSON
_size	_source 元字段占用的字节数，通过 mapper-size 插件提供

表 3-14　索引元字段

参数	说明
_all	索引所有字段的值
_field_names	文档中所有包含非空值的字段
_timestamp	关联文章的时间戳，可以手动指定或者自动生成
_ttl	定义文档被自动删除之前的存活时间

表 3-15　路由元字段

参数	说明
_parent	用于在映射类型之间创建父子关系
_routing	一个自定义的路由值，路由文档到一个特定的分片

_all

是一个特殊的包含全部内容的字段，在一个大字符串中关联所有其他字段的值，使用空格作为分隔符。所以被分析和索引但不会被存储。使用_all字段可以对文档的值进行搜索但不必知道包含所需值的字段明。

_id字段

每个被索引的文档都关联一个_type字段和一个_id字段没有索引，它的值可以从uuid字段生产。

id字段的值可以在查询以及脚本中访问，但是在聚合或者排序的时候用，要使用_uid字段而不能用_id字段。